

# Le chiffrement à flot

**Anne Canteaut**

INRIA-projet CODES

Domaine de Voluceau

78153 Le Chesnay

`Anne.Canteaut@inria.fr`

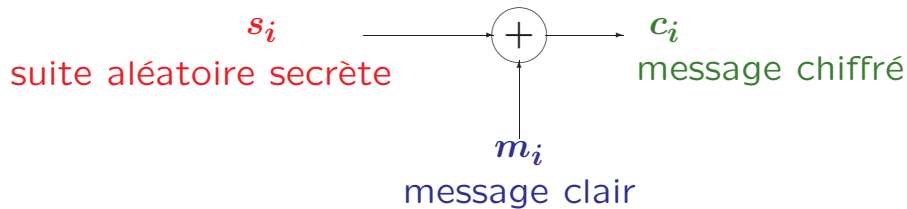
`http://www-rocq.inria.fr/codes/Anne.Canteaut/`

Ecole Jeunes Chercheurs en Algorithmique et Calcul Formel  
Montpellier, 5 avril 2005

## Plan

1. Constructions classiques
2. Attaques par corrélation (rapides)
3. Attaques algébriques
4. Exemples et perspectives

## Chiffrement de Vernam

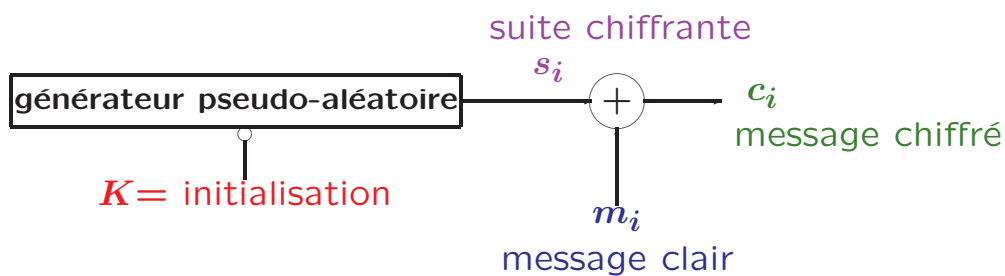


**Chiffrement parfait :** la connaissance du chiffré ne donne aucune information sur le message clair.

**Chiffrement inutilisable en pratique :** la clef secrète doit être aussi longue que le message à transmettre.

2

## Chiffrement à flot additif



**Résistance aux attaques à clair connu :**

- **attaques complètes**, qui retrouvent l'initialisation à partir de  $N$  bits de suite chiffrente ;
- **attaques par distingueur**, qui distinguent  $N$  bits de suite chiffrente d'une suite aléatoire .

Aucune de ces attaques ne doit être plus efficace que la recherche exhaustive de la clef secrète.

3

## Contextes d'utilisation

Dans deux types particuliers d'applications :

- applications logicielles qui demandent un débit de chiffrement extrêmement élevé (plus élevé que les systèmes par blocs) ;
- applications matérielles qui disposent de ressources extrêmement limitées (en mémoire, en surface, ...)

Les chiffrements à flot correspondant à des systèmes par blocs utilisés dans des modes opératoires particuliers (OFB, CTR, ...) ne répondent pas à ces demandes.

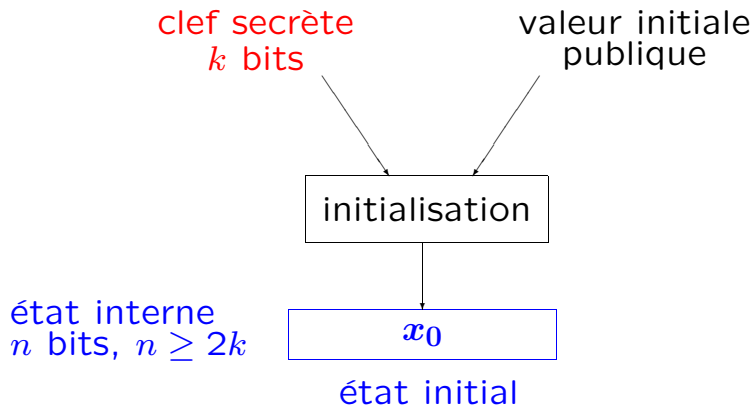
⇒ Besoin de chiffrements à flot dédiés dans ces 2 contextes applicatifs.

4

## Constructions classiques

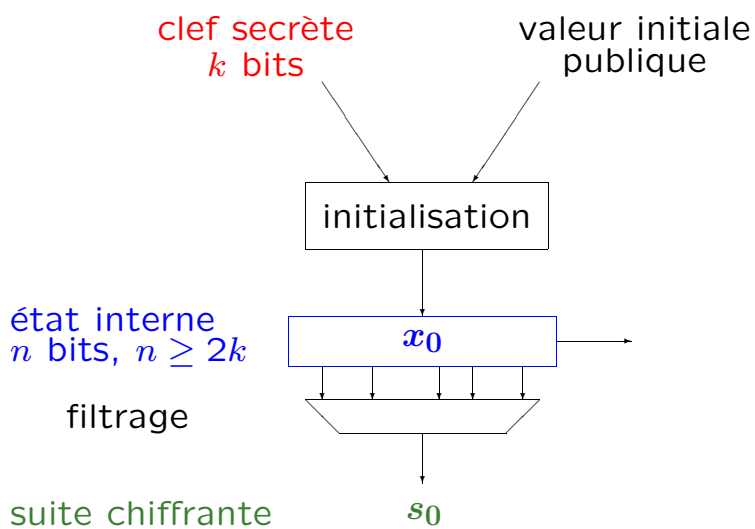
5

## Principe général de construction



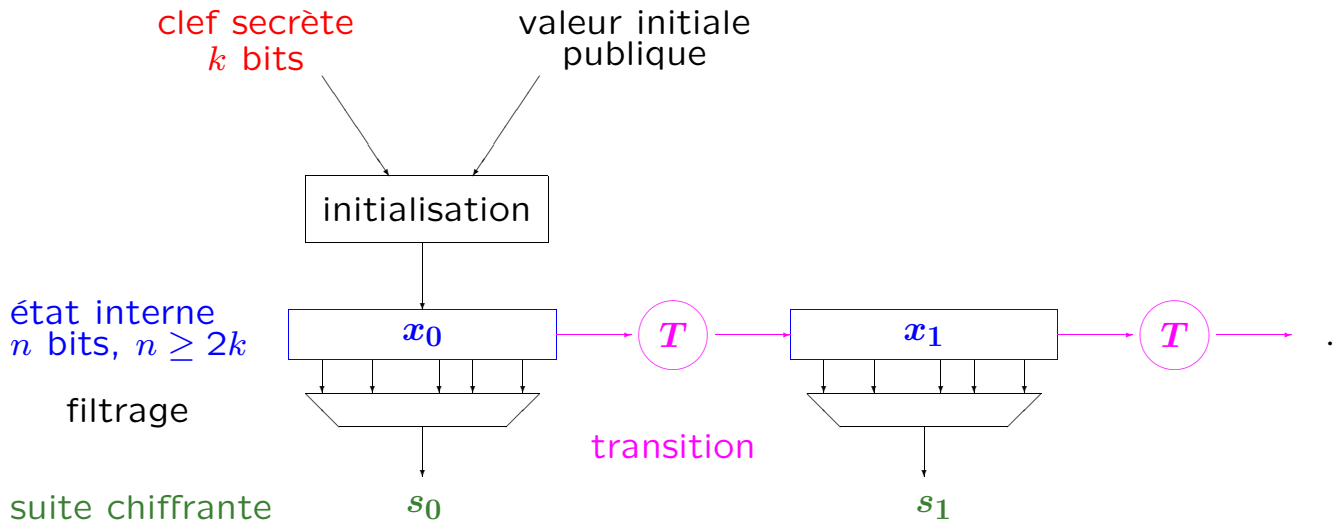
6

## Principe général de construction



7

## Principe général de construction



8

## Techniques pour accroître la sécurité

- **Horloge irrégulière** : une autre fonction détermine à chaque instant le nombre d'itérations effectuées par la fonction de transition d'état avant de produire une entrée pour la fonction de filtrage.
- **Composition** : plusieurs algorithmes à flot sont composés par addition de leurs sorties, en cascade . . .
- **Ajout de mémoire** : soit au niveau de la fonction de transition, soit au niveau de la fonction de filtrage.

9

## Fonctions de transition linéaires

### Exigences :

- garantie que la suite

$$(T^t(x_0), 0 \leq t \leq 2^n - 1)$$

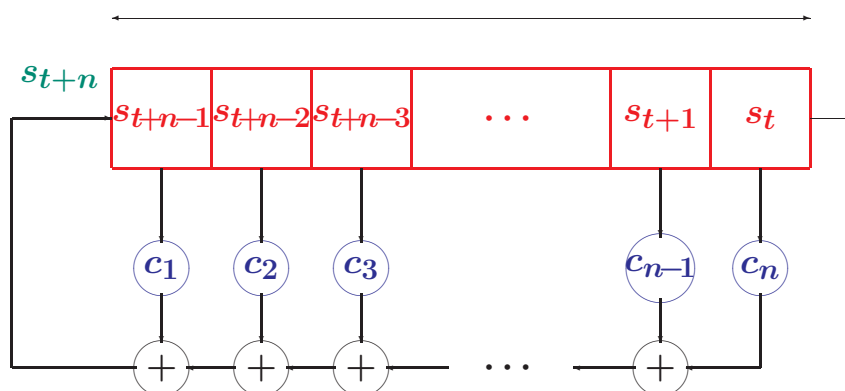
est de période maximale,  $2^n - 1$ , pour tout  $x_0 \neq 0$ .

- facilité d'implémentation.

10

### Registre à décalage à rétroaction linéaire

longueur  $n$



$$\forall t \geq n, s_t = \sum_{i=1}^n c_i s_{t-i}$$

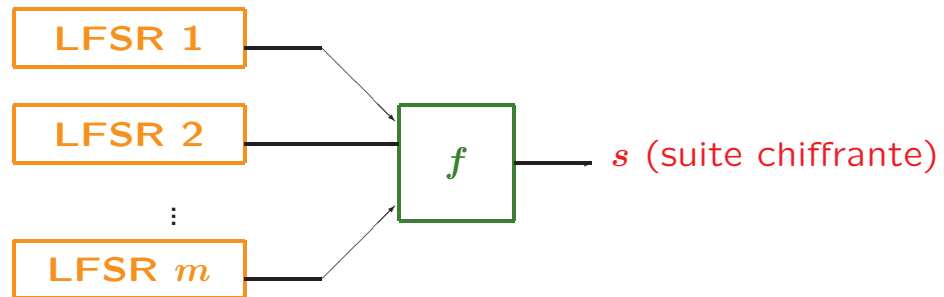
La période est égale à  $2^n - 1$  quand le polynôme de rétroaction

$$P(X) = 1 + c_1 X + c_2 X^2 + \dots + c_n X^n$$

est primitif.

11

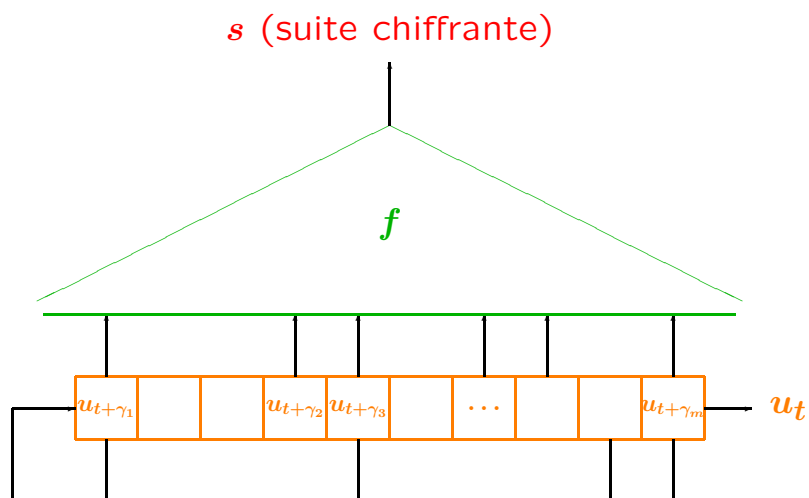
## Générateurs par combinaison de registre



où  $f$  est une fonction booléenne à  $m$  variables équilibrée.

12

## Registres filtrés



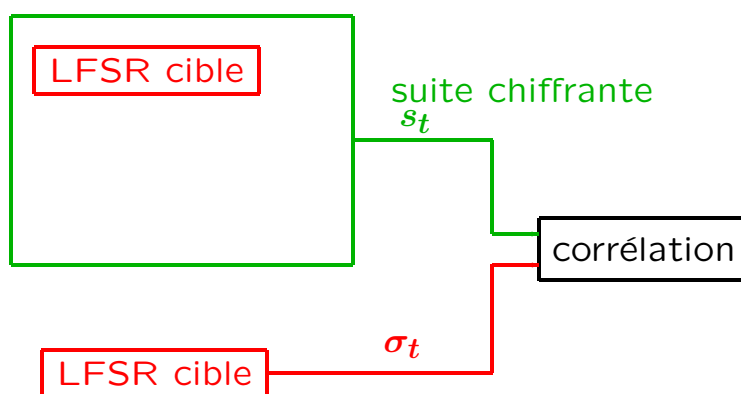
$$\forall t \geq 0, s_t = f(u_{t+\gamma_1}, u_{t+\gamma_2}, \dots, u_{t+\gamma_m})$$

13

## Attaques par corrélation (rapides)

14

### Attaque par corrélation [Siegenthaler 85]



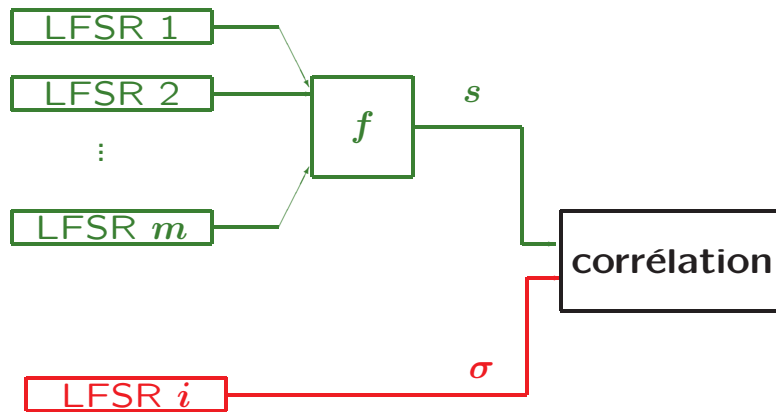
$$\text{où } p = \Pr[s_t \neq \sigma_t] \neq \frac{1}{2}.$$

#### Problème :

Retrouver l'état initial du LFSR cible à partir de la connaissance de certains bits de la suite chiffrante.

15

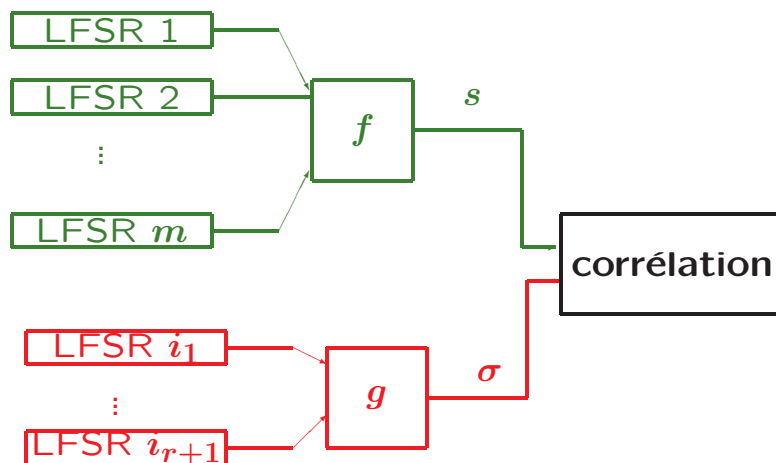
## Attaque par corrélation sur une combinaison de LFSRs



avec  $P[f(x_1, \dots, x_m) \neq x_i] = P[s_t \neq \sigma_t] \neq \frac{1}{2}$ .

16

## Généralisation de l'attaque de Siegenthaler



avec  $P[f(x_1, \dots, x_m) \neq g(x_{i_1}, \dots, x_{i_{r+1}})] < \frac{1}{2}$ .

17

## Choix de la fonction $g$

Soit  $f$  une fonction à  $n$  variables, équilibrée.

Son ordre de non-corrélation est le plus grand nombre  $r$  tel que

$$P[f(x_1, \dots, x_m) \neq g(x_{i_1}, \dots, x_{i_r})] = \frac{1}{2}$$

pour tout ensemble  $\{i_1, \dots, i_r\} \subset \{1, \dots, m\}$  et pour toute fonction  $g$  à  $r$  variables.

### Théorème [Canteaut-Trabbia 00] [Zhang 00]

Il existe un sous-ensemble de  $r + 1$  variables,  $\{x_{i_1}, \dots, x_{i_{r+1}}\}$  et une fonction  $g$  à  $(r + 1)$  variables tels que

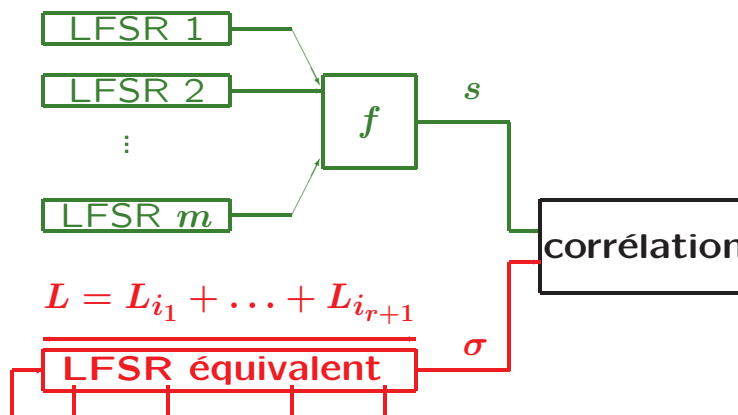
$$p_g = P[f(x_1, \dots, x_m) \neq g(x_{i_1}, \dots, x_{i_{r+1}})] < \frac{1}{2}.$$

La probabilité  $p_g$  est minimale pour la fonction affine

$$g(x_{i_1}, \dots, x_{i_{r+1}}) = \sum_{j=1}^{r+1} x_{i_j} + \varepsilon, \varepsilon \in \{0, 1\}.$$

18

## Attaque par corrélation généralisée



avec  $P[s_t \neq \sigma_t] = P[f(x_1, \dots, x_m) \neq x_{i_1} + \dots + x_{i_{r+1}}] < \frac{1}{2}.$

19

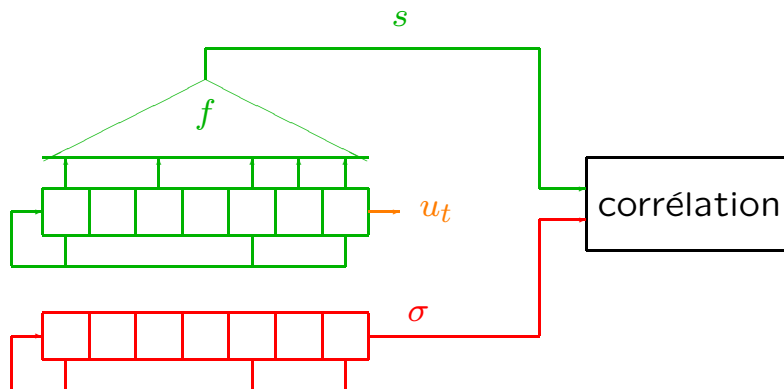
## Attaque par corrélation d'un registre filtré

Soit  $\alpha \in \mathbb{F}_2^m$  qui minimise

$$p_\alpha = P[f(x_1, \dots, x_m) \neq \alpha \cdot (x_1, \dots, x_m)] = P[s_t \neq \sigma_t]$$

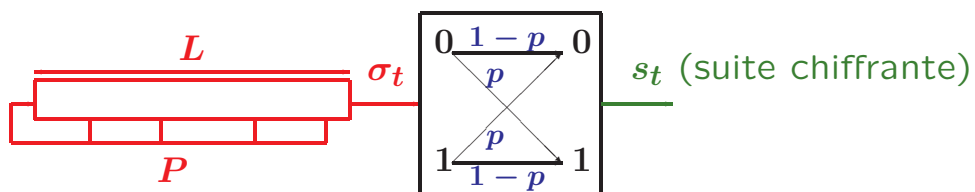
où  $\sigma_t = \alpha \cdot (u_{t+\gamma_1}, \dots, u_{t+\gamma_m})$ .

La suite  $\sigma$  est produite par un LFSR identique au LFSR d'origine, mais initialisé par les  $\alpha \cdot (u_{t+\gamma_1}, \dots, u_{t+\gamma_m})$ ,  $0 \leq t < n$ .



20

## Attaque par corrélation rapide [Meier-Staffelbach 88]



probabilité d'erreur :

$$p = P[s_t \neq \sigma_t] < \frac{1}{2}$$

$(\sigma_t)_{t < N}$  est un mot du code linéaire  $[N, L]$  défini par  $P(X)$ .

21

## Attaque par corrélation rapide et décodage

$$\forall t, \sum_{i=0}^L c_i \sigma_{t-i} = 0 .$$

Code linéaire de longueur  $N$  et de dimension  $L$  :

$$(\sigma_0, \dots, \sigma_{L-1}) \begin{bmatrix} 1 & 0 & \dots & 0 & c_L & \dots \\ 0 & 1 & \dots & 0 & c_{L-1} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & c_1 & \dots \end{bmatrix} = (\sigma_0, \dots, \dots, \sigma_{N-1})$$

où la colonne  $t$  de la matrice est donnée par

$$\sum_{i=0}^{L-1} g_{i,t} X^i = X^t \bmod P^*(X) \text{ où } P^*(X) = \sum_{i=0}^L c_{L-i} X^i .$$

22

## Décodage à maximum de vraisemblance [Siegenthaler 85]

**Mot reçu :**

$N$  bits de suite chiffrante  $s$ , probabilité d'erreur :  $p$ .

**Algorithme :**

Pour chacun des  $2^L$  états initiaux possibles  $\sigma = (\sigma_0, \dots, \sigma_{L-1})$

Calculer la distance entre  $\sigma G$  et  $s$ :

$$d_H(s, \sigma G)$$

Retourner la valeur de  $\sigma$  qui minimise

$$d_H(s, \sigma G)$$

**Complexité**

$N 2^L$  ou  $L 2^L$  avec une FFT.

23

## Nombre minimal de bits de suite chiffrante à connaître

### Théorème de codage de canal [Shannon 48]

Soit  $C(p)$  est la capacité du canal binaire symétrique avec probabilité d'erreur  $p = \frac{1}{2} - \epsilon$ ,

$$C(p) = 1 + p \log_2 p + (1 - p) \log_2(1 - p) \simeq \frac{2\epsilon^2}{\ln(2)} .$$

Il faut connaître  $N$  bits de la suite  $s$  avec

$$N \geq \frac{L}{C(p)} \simeq \frac{\ln(2)L}{2\epsilon^2}$$

⇒ trouver un algorithme de décodage efficace pour ce code pour lequel le nombre de bits  $N$  de la suite  $s$  nécessaires soit le plus proche possible de la borne de Shannon.

24

### Codes correcteurs et algorithmes de décodage utilisés

- Décodage à maximum de vraisemblance du code initial ;
- Code convolutif et algo. de Viterbi [Johansson-Jönsson 99] ;
- Turbo-code [Johansson-Jönsson 99] ;
- Codes à matrice de parité creuse et algorithme de Gallager [Meier-Staffelbach 88], [Canteaut-Trabbia 00], [Mihaljevic - Fossorier- Imai 00] ;
- Décodage à maximum de vraisemblance d'un code de dimension plus petite [Chepyshov - Johansson - Smeets 00].
- Algorithme de Sudan pour reconstruire un polynôme linéaire [Johansson-Jönsson 00].

25

## Utilisation d'équations de parité creuse et décodage itératif

[Meier-Staffelbach 88], [Canteaut-Trabaccia 00]

- On cherche des équations creuses vérifiées par  $\sigma$ .

Si  $P(X)$  divise  $1 + X^a + X^b$ , alors

$$\forall t, \sigma_t + \sigma_{t-a} + \sigma_{t-b} = 0 .$$

- On décode le mot reçu  $(s_t)_{t < N}$  relativement à un code à matrice de parité creuse avec un algorithme itératif.

26

### Phase de précalcul

Trouver toutes les équations linéaires contenant  $d$  bits de  $(\sigma_t)_{t < N}$   
 $\implies$  Multiples de  $P$  de poids  $d$  et de degré au plus  $N$ .

$$m(d) \simeq \frac{N^{d-1}}{(d-1)! 2^{\deg(P)}} \text{ équations par bit .}$$

Cette approximation ne dépend pas du poids de  $P$ .

$$P_1 = 1 + X^3 + X^{17}$$

$$P_2 = 1 + X^2 + X^4 + X^5 + X^6 + X^8 + X^9 + X^{10} + X^{11} + X^{13} + X^{14} + X^{15} + X^{17}$$

$N$	3000	4000	5000	6000	7000	8000
$m_3$ pour $P_1$	38	61	95	131	183	238
$m_3$ pour $P_2$	36	67	95	127	185	243
approximation	34	61	95	137	187	244

27

## Phase de décodage (belief propagation)

Observation :

$$Obs(\sigma_t) = Pr[\sigma_t = 1|s] = \begin{cases} p & \text{si } s_t = 0 \\ 1 - p & \text{si } s_t = 1 \end{cases}$$

Information extrinsèque sur  $\sigma_t$  dans sa  $m$ -ième équation:

Pour la  $m$ -ième équation contenant  $\sigma_t$ :  $\sigma_t = \bigoplus_{j \in J_m} \sigma_j$ .

$$Ext_m(\sigma_t) = Pr[\bigoplus_{j \in J_m} \sigma_j = 1|s]$$

Probabilité a posteriori :

$$APP(\sigma_t) \propto Obs(\sigma_t) \prod_{m=1}^{m_d} Ext_m(\sigma_t)$$

L'information extrinsèque  $Ext_m(\sigma_t)$  est mise à jour en utilisant les **APPs partielles** (i.e., les APPs en excluant l'information extrinsèque donnée par l'équation  $m$ ).

28

## Complexité de la phase de décodage [Leveiller 04]

Belief propagation original [Gallager 62]

Complexité en temps :  $\mathcal{O}(\text{nb}_{\text{ite}} N m_d^2)$  Mémoire :  $\mathcal{O}(N m_d)$ .

Approximations :

- Utilisation des **APP totales au lieu des APPs partielles**

Complexité en temps :  $\mathcal{O}(\text{nb}_{\text{ite}} N m_d)$  Mémoire :  $\mathcal{O}(N)$ .

- Utilisation d'une **approximation pour calculer les informations extrinsèques**

$\implies$  réduction de la constante dans la complexité en temps, mais dégradation des performances.

29

## Complexité de l'attaque par corrélation rapide

Pour une probabilité d'erreur  $p = \frac{1}{2} - \varepsilon$ :

Nombre d'équations de poids  $d$  nécessaires au décodage :

$$m_d \geq \frac{2 \ln 2}{(2\varepsilon)^{2d-4}} \text{ équations par bit.}$$

Nombre de bits de suite chiffrante :

$$N \propto \left( \frac{1}{2\varepsilon} \right)^{\frac{2(d-2)}{d-1}} 2^{\frac{L}{d-1}} .$$

$$\text{Précalcul} \simeq \frac{N^{d-2}}{(d-2)!} \quad \text{Décodage} \propto \left( \frac{1}{2\varepsilon} \right)^{\frac{2d(d-2)}{d-1}} 2^{\frac{L}{d-1}} .$$

Pour  $d = 4$  :

$L = 40$ ,  $p = 0.44$ ,  $N = 400\,000$  bits de suite chiffrante.

Précalcul : 9 h, décodage : 1.5 h.

$L = 60$ ,  $p = 0.4$ ,  $N = 900\,000$  bits de suite chiffrante,  $2^{38}$  opérations.

30

## Attaques algébriques

31

## Générateurs avec une fonction de transition linéaire

- L'état initial est identifié à la clef secrète :

$$\text{état initial} = k_0, \dots, k_{n-1} \in \mathbb{F}_2^n$$

- Fonction de transition linéaire sur  $\mathbb{F}_2^n$ :

$$x_t = L^t(k_0, \dots, k_{n-1})$$

- Fonction de filtrage de  $\mathbb{F}_2^n$  dans  $\mathbb{F}_2$ :

$$s_t = f(x_t) = f \circ L^t(k_0, \dots, k_{n-1})$$

**Problème.** Retrouver l'état initial  $k_0, \dots, k_{n-1}$  à partir de la connaissance de  $N$  bits de suite chiffrante  $s_0, s_1, \dots, s_{N-1}$ .

32

## Attaque de Shannon

**Idée fondatrice.**

$$\begin{cases} s_0 = f(k_0, \dots, k_{n-1}) \\ s_1 = f \circ L(k_0, \dots, k_{n-1}) \\ s_t = f \circ L^t(k_0, \dots, k_{n-1}) \end{cases}$$

Systeme d'équations à  $n$  variables de degré  $\deg(f)$  .

$\implies$  Résolution par linéarisation

$$\sum_{i=1}^d \binom{n}{i} \simeq n^d \text{ bits de suite chiffrante}$$

Complexité :  $n^{3d}$  opérations .

33

## Attaques algébriques [Courtois-Meier 03]

Soit  $AN(f) = \{g, g(x)f(x) = 0 \text{ pour tout } x \in \mathbb{F}_2^n\}$ .

Soit  $g \in AN(f)$ , i.e.,  $g(x)f(x) = 0$  pour tout  $x$ .

$$g(x_t)f(x_t) = g(x_t)s_t = 0$$

$$\implies g \circ L^t(k_0, \dots, k_{n-1}) = 0 \text{ if } s_t = 1 .$$

Soit  $h \in AN(1+f)$ , i.e,  $h(x)(1+f(x)) = 0$  pour tout  $x \in \mathbb{F}_2^n$ .

$$h(x_t)(1+f(x_t)) = h(x_t)(1+s_t) = 0$$

$$\implies h \circ L^t(k_0, \dots, k_{n-1}) = 0 \text{ if } s_t = 0 .$$

Système d'équations à  $n$  variables de degré

$$d = \min\{\deg(g), g \in AN(f) \cup AN(1+f), g \neq 0\} .$$

34

## Complexité de l'attaque

$n$  = taille de l'état interne

$AI(f)$  = immunité algébrique de la fonction de filtrage  $f$   
 $= \min\{\deg(g), g \in AN(f) \cup AN(1+f), g \neq 0\}$ .

Nombre d'opérations :

$$\left( \sum_{i=1}^{AI(f)} \binom{n}{i} \right)^\omega \simeq n^{AI(f)\omega} \text{ où } \omega \simeq 2.37$$

Critère de sécurité :

Pour  $n = 2k$  où  $k$  est la taille de la clef, il faut  $(2k)^{AI(f)\omega} \geq 2^k$   
i.e.,

$$AI(f) \geq 0.42 \left[ \frac{k}{1 + \log_2 k} \right]$$

**Exemple.**  $k = 128$  bits,  $n = 256$  bits.

$$\longrightarrow AI(f) \geq 7 .$$

35

## Autres techniques pour résoudre le système

### Linéarisation

- linéarisation simple
- XL [Courtois-Klimov-Patarin-Shamir 00]

### Algorithmes de bases de Gröbner

- F4 [Faugère 99]: plus efficace que XL [Ars et al 04][Diem 04]
- F5 [Faugère 02]: strictement plus efficace que les précédents.

### Techniques ad hoc

- XSL [Courtois-Pieprzyk 02] (implémentation ?)

**Problème ouvert.** *Le système obtenu dans l'attaque est-il un système générique ?*

36

### Existence de $g \in AN(f)$ avec $\deg g \leq d$

$x$  tels que  $f(x) = 1$  [wt(f)]

1	$RM^f(d, n)$	tous les monômes de degré $\leq d$ [ $\sum_{i=0}^d \binom{n}{i}$ ]
$x_1$		
⋮		
$x_n$		
$x_1 x_2$		
⋮		
$x_{n-1} x_n$		

$$\dim\{g \in AN(f), \deg g \leq d\} = \sum_{i=0}^d \binom{n}{i} - \text{rang} \left( RM^f(d, n) \right) .$$

**Proposition** Il existe  $g \neq 0$  dans  $AN(f)$  avec  $\deg g \leq d$  si

$$wt(f) < \sum_{i=0}^d \binom{n}{i} .$$

37

**Bornes sur l'immunité algébrique**  
[Courtois-Meier 03][Dalai-Gupta-Maitra 04]

**Proposition**

Soit  $f$  une fonction booléenne à  $n$  variables. Si  $AI(f) \geq d$ , alors

$$\sum_{i=0}^d \binom{n}{i} \leq wt(f) \leq 2^n - \sum_{i=0}^d \binom{n}{i}$$

**Corollaire** Pour toute  $f$  à  $n$  variables,

$$AI(f) \leq \left\lceil \frac{n}{2} \right\rceil .$$

De plus, si  $f$  a une AI optimale, alors

- si  $n$  est impair,  $wt(f) = 2^{n-1}$
- si  $n$  est pair,

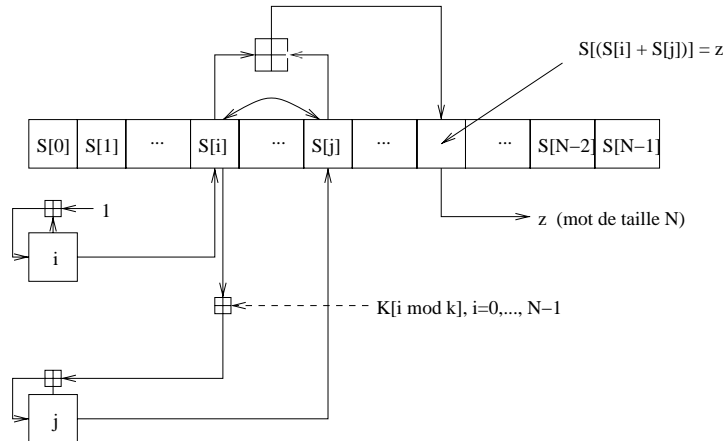
$$2^{n-1} - \frac{1}{2} \binom{n}{n/2} \leq wt(f) \leq 2^{n-1} + \frac{1}{2} \binom{n}{n/2} .$$

38

## Exemples et perspectives

39

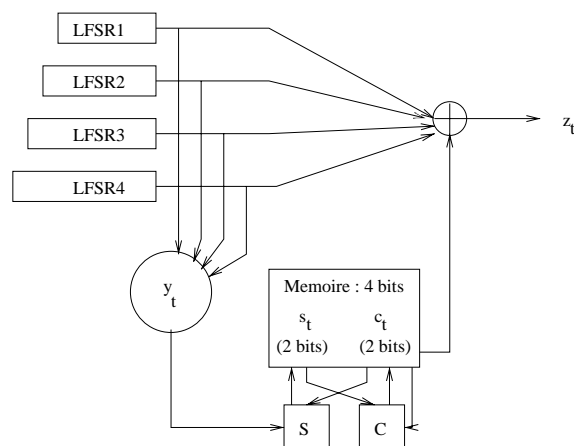
## RC4 [Rivest 87]



- **Taille de clef** : entre 40 et 1024 bits. (état = 256 octets).
- **Performances** : 7.3 cycles par octet sur un Pentium III.
- **Utilisation** : dans SSL et le WEP
- **Attaques** : pas d'attaques génériques, mais des propriétés particulières, des biais linéaires ou statistiques,...

40

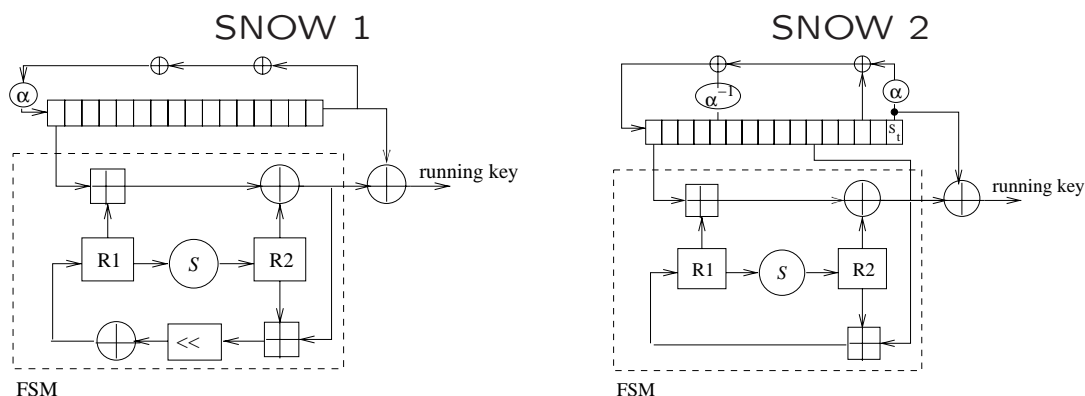
## E0



- **Taille de clef** : 64 ou 128 bits (états des LFSRs = 128 bits).
- **Utilisation** : dans Bluetooth (protocole Wireless LAN)
- **Attaques** : attaque par corrélation rapide [Lu Vaudenay 04] :  $2^{37}$  précalculs,  $2^{39}$  calculs et  $2^{39}$  bits de sortie.

41

## SNOW [Ekdahl Johansson 02]



- clé de taille 128 ou 256 bits, utilisation de mots de 32 bits avec une machine à états finis.
- **Performances** : 1 octet pour 8 cycles sur un P III
- **Attaques** : aucune sur SNOW-2, une attaque par distingueur sur SNOW-1.

42

### Constructions utilisant une fonction de transition non-linéaire

**Problème** : Trouver une fonction de transition rapide, non-linéaire (pour éviter les attaques précédentes), de période élevée.

### Les T-fonctions [Klimov Shamir 02]

$$T : (x_0, \dots, x_{n-1}) \mapsto (y_0, \dots, y_{n-1})$$

telle que  $y_k$  dépend uniquement de  $(x_0, \dots, x_k)$ .

Il existe des T-fonctions non linéaires composées de fonctions élémentaires ( $\oplus$ , addition modulo  $2^n$ , multiplication sur les entiers,...) de période maximale.

$$x \mapsto x + (x^2 \vee 5) \pmod{2^n}$$

**Problème** : comment choisir la fonction de filtrage ?

43

## Perspectives

### Evaluation de la sécurité des systèmes existants

- amélioration générale des attaques connues,
- attaques dédiées.

### Conception d'un nouveau chiffrement à flot

- recherche de critères permettant de garantir la résistance aux attaques connues,
- étude des systèmes utilisant des fonctions de transition non-linéaires.